

SPECIFICATIONS

DTBOOT

DECsystem-10 DECTAPE BOOTSTRAP

**Date: 14-Jun-73
File: DTBOOT.RNO
Edition: 4**

This document describes the software as of Version 3.

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation.

Actual distribution of the software described in this specification will be subject to terms and conditions to be announced at some future date by Digital Equipment Corporation.

DEC assumes no responsibility for the use or reliability of its software on equipment which is not supplied by DEC.

This software is furnished to purchaser under a license to use on a single computer system and can be copied (with inclusion of DEC's copyright notice) only for use in such system, except as may otherwise be provided in writing by DEC.

CONTENTS

- 1.0 Purpose of DTBOOT**
- 2.0 Commands to DTBOOT**
- 3.0 Tape Format**
- 4.0 Comparison with TENDMP**
- 5.0 Compatibility with DECsystem-10 Monitor**
- 6.0 Assembly and READIN of DTBOOT**

Appendix - DTBOOT Summary for Operators

1.0 PURPOSE OF DTBOOT

DTBOOT is a bootstrap program intended to transfer programs between DECsystem-10 core storage and standard format DECsystem-10 DECTape files, and to perform other utility functions with such DECTapes.

DTBOOT requires a DECsystem-10 central processor and core memory, the DECsystem-10's console typewriter, a TD10 DECTape control, and at least one DECTape transport. DTBOOT can be assembled for any memory size.

DTBOOT is intended to load and run executive mode programs (monitors, diagnostics, etc.) from DECTape.

The following functions are available:

1. Clear core, load a program, and start it.
2. Load a program without clearing other core areas.
3. Clear core, load a program, but do not start it.
4. Type out a short form of the DECTape's directory.
5. Delete a file from the DECTape.
6. Initialize an empty tape directory, either retaining its existing tape identifier word or supplying a new one.
7. Declare a starting address to be associated with a program to be dumped or started.
8. Dump core onto a DECTape file.
9. Go to the current starting address.

A simple command string, with handy defaults, is typed by the operator to control DTBOOT.

2.0 COMMANDS TO DTBOOT

The commands for DTBOOT are best described by showing the longest form and then showing the simplifications provided.

The full command to load the file TOPS10.SAV from DECTape unit DTA3 is:

```
DTA3:TOPS10.SAV/L
```

The command terminator may be carriage return, or any other control character or ALTMODE.

The unit specifier DTA3: is actually examined only for numeric characters, so it can be abbreviated to simply 3:. If a series of commands is performed addressing the same unit, the unit specifier can be omitted on successive commands (unless an error has occurred, causing a BELL typeout).

The file extension is assumed to be .SAV unless specified otherwise, and can therefore be omitted in the usual case. A blank extension can be specified by typing just the period; e.g., "3:NAME./L".

The filename can be omitted, and is then defaulted according to the following rules:

1. For the tape-writing command, the file name CRASH is assumed.
2. For tape reading, if exactly one file exists on the tape with the extension .SAV, that file's name is assumed. Otherwise, the file name SYSTEM is assumed.

The particular action to be performed by DTBOOT is selected by a switch, which is an octal number or single letter that follows a slash (/) at the end of the command. In the first example above, the L is the command switch, meaning Load the program into core from tape.

The most common function of DTBOOT is to load and start a program (e.g., the timesharing system), and this action is the default if no switch is typed at all.

The non-blank switches, and the actions they cause, are as follows:

- /number Set the starting address to the octal number.
- /D Dump core on tape with the current starting address.
- /F Type the file directory.
- /G Go to starting address.
- /K Kill (delete) the specified file from the tape.
- /L Clear core, load the program from tape, and set the starting address from the tape.

- /M Same as /L, but do not clear core first, i.e..merge the file into core.
- /Z Zero the directory of the tape. To set the tape identifier to TAPE99, for instance, type 3:↑TAPE99/Z. The uparrow precedes the tape identifier, as in FILEX.

Obviously, not all of these commands require all arguments. For example, a filename would be ignored with the /Z command.

3.0 TAPE FORMAT

DECTapes read and written by DTBOOT are entirely compatible with the DECsystem-10 monitor, and are fully described in the DECsystem-10 Software Notebooks. This section is provided as a summary of that material.

DECTape directory information occupies block 144 (octal) of the tape. Words 0 through 122 contain 1105 bytes describing the allocation of blocks 1 through 1101 of the tape. Blocks 1, 2, and 144 are allocated as overhead, blocks 1102 through 1105 are allocated as non-existent. The remaining blocks contain file information (or are free). Words 123 through 150 of the directory contain filenames. Words 151 through 176 contain the corresponding file extensions and dates. (Date fields of files written by DTBOOT are zero.) Word 177 contains the (optional) tape identifier.

In each data block, word zero is the link word, allocated as follows: bits 00-17 are the link to the next data block of the file (zero denoting no next block); bits 18-27 contain the block number of the first data block of this file, and bits 29-35 contain a word count of the data words in the remainder of this block.

A saved core image, as written by DTBOOT (or TOPS10), consists of blocks of data preceded by a pointer word (minus-count in the left half, address-1 in the right half). A positive word in the place of a pointer word signifies the end of the core-image. The right half of this word is the starting address of the saved program. The left half usually contains the machine instruction JRST which is, in fact, a positive number. This is the method DTBOOT uses.

Blocks of a given file may be in any order on the tape. When written by DTBOOT (and the SAVE command of TOPS10), the allocation is as follows: The lowest numbered block available for use is allocated first. (This was a requirement of TENDMP, but is optional for DTBOOT.)

Succeeding blocks are allocated with at least one intervening block (whether free or otherwise used) between successive blocks of a file. If required, the same procedure is used to allocate blocks in the reverse direction, starting at block 1101, should the file be long enough to reach that end of the tape, and any skipped blocks become used on the reverse pass. DTBOOT is capable of efficiently reading and writing files allocated in this manner.

4.0 COMPARISON WITH TENDMP

It is intended that DTBOOT supersede TENDMP. The only advantages TENDMP holds over DTBOOT are that TENDMP is smaller than DTBOOT, and that TENDMP may be called as a subroutine, without typed commands. DTBOOT lacks this facility. TENDMP will remain available for cases where these advantages are desired.

The advantages DTBOOT holds over TENDMP include the following:

1. First and most importantly, the ability to read .SAV files written by TOPS10 which are long enough to "turn around", i.e., which have allocation in both forward and backward direction.
2. Writing files which can be read efficiently by TOPS10, without frequent tape rocking due to allocation of consecutive blocks by TENDMP. (In fact, DTBOOT has the same difficulty reading TENDMP files as TOPS10 does.)
3. More complete DECTape control handling; the ability to turn around from end-zone flags.
4. More nearly standard command strings.
5. Useful defaults.
6. Ability to set tape-identifier words.

5.0 COMPATIBILITY WITH DECsystem-10 MONITOR

There are no known incompatibilities with TOPS10, aside from the failure of DTBOOT to provide dates on output files.

6.0 ASSEMBLY AND READIN OF DTBOOT

As with TENDMP and BOOTS, DTBOOT is assembled by MACRO-10 from a single source file onto a paper tape.

DTBOOT resides at the top of user core, and thus must be assembled for the proper memory size. The default is for a 64K system. The parameter CORE can be set to the number of words of memory to produce another size of DTBOOT.

To assemble 64K DTBOOT, type the MACRO command string:

```
*PTP:+dev:DTBOOT
```

where dev is the device containing the source file of DTBOOT.

To assemble 48K DTBOOT, for example, type:

```
*PTP:+TTY:,dev:DTBOOT
CORE=140000 ;ASSEMBLE FOR 48K
+Z
+Z
```

DTBOOT is in RIM10B format when on paper tape, and is loaded via READIN at the PDP-10 console.

The starting address of DTBOOT is 1000 octal less than CORE. Thus, 64K DTBOOT starts at 177000.

There are three blocks (0, 1, 2) available for a bootstrap at the beginning of a PDP-10 DECTape. These 600 octal words, minus a READIN pointer and a start word, leave 576 octal for the bootstrap. At present DTBOOT is 575 words long, and is intended to be stored on DECTape in this area. Instructions for putting DTBOOT on the front of DECTape are as follows: (Refer to files COPY.OPR and COPY.DOC for additional information.)

```
.ASSIGN DSK PTR
.RENAME BSLDR.REL=DTBOOT.REL
.R COPY
*/L ;load bootstrap loader
*+C
.SAVE DSK COPY ;save on disk
START
*DTAn:/T ;write on front of DECTape
TYPE CORE BANK OR OFFSET FOR DTBOOT
64K ;answer size of core bank
or OFFSET, ex., 64K=1770000
```

APPENDIX: DTBOOT SUMMARY FOR OPERATORS

DTBOOT is loaded from paper tape via the READIN switch on the PDP-10. DTBOOT starts at 77000 (32K), 137000 (48K), 177000 (64K), 237000 (80K), etc.

COMMANDS

DTA3:name.ext/L or 3:name.ext/L

SWITCHES

none	= load and start program
/number	= set starting address to number (octal)
/D	= dump core and starting address on tape
/F	= type file directory
/G	= go to starting address
/K	= kill (delete) file
/L	= load, set starting address, do NOT start
/M	= merge (load without clearing core first)
/Z	= zero the tape directory (3:tapeid/+Z sets ID)

DEFAULTS

dev:, initially DTA0:. Otherwise, same as on previous command (unless the bell was rung).

name=CRASH for /D. Otherwise, if only one .SAV file on tape, use its name. If more than one .SAV file, use SYSTEM, unless specified.

ext = .SAV unless specified.

ERRORS

If an error occurs, the bell rings. The possible errors include:

1. Tape unit not in remote.
2. Not exactly one unit dialed as dev:
3. Parity error on tape.
4. Requested file not in directory. (Type /F to list directory)
5. Tape filled up before writing is completed.
6. Invalid command string.